So in the last lesson we've successfully tapped into the Open Weather Map's data servers and we've gotten back the data for the current weather.

Now this weather data is formatted in what's called a JSON format.

And what you'll find is that most API providers are split into two camps.

They'll either provide you the data in the XML format, the extensible markup language format or something called a JSON format. And some API providers even give you a choice as to which type of response you would like. And Open Weather Map is one of those types of providers.

So you can see here that we're getting our data back in our JSON format. But we can also say that we want the XML mode to get the same data back in the XML style. Now previously, we've already seen what an XML file looks like when went into the Android manifest and the iOS plist.

So when we take a look at the app, src, main and we open up our Android manifest, you can see that this is styled in the SML format. And XML looks very similar to HTML if you've ever used that. It's essentially a whole bunch of key value pairs.

So the name of the key is the tag and there's an opening tag and a closing tag. And you can tell that it's a closing tag because it has a forward slash before the key. In between those tags is the actual value of the key.

So the key could be something like temperature forward slash temperature in the middle it could be 32 or whatever value it might be. Now JSON however stands for JavaScript Object Notation, so it's a format that looks extremely similar to how you would create objects in JavaScript. And the way that it's styled looks something like this. Every key value pair is enclosed in a set of curly braces.

And the first part is the key,then there's a colon and then there's the value. And this looks very similar to the maps that we have in Dart as well. Now this is our weather data in a JSON format.

So for example in this case the coordinates here is the first key and the value is everything that comes afterwards.

So it's the lon and the lat.

Now you can see that the value for the coordinates itself has a key value pair, where lon is the key and -0.13 is the value.

Now as a human, it's actually quite hard to be able to read this without having it formatted. So a really useful tool is something like a JSON viewer.

So this is a Chrome plugin called JSON Viewer Awesome and it's free to download and add to your chrome browser.

And now if we go back to where we had our JSON data in the response and we hit refresh, you can see that JSON viewer has now formatted all of that data into a format that's much much easier to read as a human.

So essentially we've got a single JSON, which contains 12 items.

So one of those is coordinates,one is weather, one is base,one is main.

But each of these object then have their own objects inside it.

So the coord contains the lon and the lat, the weather contains one item and the item contains an ID for the weather condition, the weather name.

So it's currently drizzling in London as always.

And it's got a description and you can also view it as a chart.

So a whole bunch of nodes.

So for example response goes the coordinates, goes lon and goes lat.

And when you hover over them, you can view the values like this.

So the way that I would view a JSON is it's almost like those flat pack furniture things where if you were to create say a map in Dart, you might write var wardrobe =you open up your curly braces and then you have some key and value pairs.

So our wardrobe has two doors, has two drawers and it has the color red.

Now if we were to transport this data across the Internet, that's actually quite a lot of characters.

How can we make it more flat pack, more like our IKEA furniture?

Well we could probably just have those key and value pairs and take out the rest of the grammar or the syntax.

So it might look something like this.

And once we've transported this data across the Internet, keeping it as small as possible.

Once it lands on the floor of the location that it needs to be, so in our app for example,well then we can unpack it and assemble it back into its original form.

So now if I was to try and unpack the data from our response for the weather, if I for example wanted to tap into this value so the value of the longitude, then what I would have to do is we have to use the help of a Dart package called dart:convert.

And once we import dart:convert, then we can start using a method that comes from that package called jsonDecode.

And here we can pass in a source,so a bit of JSON data, and it'll will try to decode it depending on what values we want.

So for example if I wanted to unpack my JSON here for my weather data and I wanted to access the value of the longitude here, then I would first take my source which is going to be the data that I get back as the response body which we know to be JSON formatted, and then I could provide a key. The first key to get to this longitude is the coord key.

So I have to step into here and then I have to make my step into lon before I can tap into this value.

So the first key is going to be the string coord, and it's really important that you spell it exactly the same way that you see in the data,otherwise it won't work.

Now after that stage I now am inside here.

And in order to get my actual value, I have to give it the second key which is lon.

So I'm going to add another set of square brackets and add the word lon, so spelt exactly the same way as I see here.

So now this is going to be equal to the value of the longitude, which is 139.01.

And I can now save that into a variable, let's call that the longitude.

And now if I go ahead and print my longitude instead of data, we can see what we actually get from unpacking our JSON data.

So let's hit save and let's check out our console.

You can see that I'm getting 139.01,exactly the same as here.

So I had to step through two keys in order to get to that value.

Now if I wanted to get to the weather condition for example, if I wanted to get the description here,now here it's a little bit more tricky.

The first step is very easy. Let's create a new variable, let's call it the weatherDescription. And let's set it to equal, again we're going to use JSON decode to decode our data and then we have to pass it some keys.

So how do we reach this description? Well the first step is through the key weather, so let's add that in a square bracket.

Now the next key is not what you might expect.

So notice that here, we have one arrow so that stored data, our second arrow which is our coord and our third one is simply the key longitude.

Here we have are data then our weather, then we have something here which doesn't actually have a name.

So when you collapse our weather, you can see that it's got a list that only contains one item.

So to access the first item in a list what do we do?

Well we have to use the index of that list for the first value. And we know that lists start counting from zero.

So here we have to add a square bracket and add zero to be able to go into this tree right here. Because notice that this ID is not directly inside the weather, unlike the longitude which is directly inside the coordinate.

So now by adding that zero we're now inside this tree and we can now add our final key which is the description.

So making sure that I've actually spelt it right description.

This should now be equal to the weather description, so let's check it by printing it here. Let's print the weather description and if we hit saver, let's check our console. We're getting clear sky printed in here which comes from right here.

Now if that seems really really painful, it's because it's actually the hard way of doing things.

And by using something like JSON Viewer Awesome, there's actually a much easier way.

So for example if I wanted to get the main value here, I can click on it, select it and then in the gutter here I can click on copy path.

So now I can go ahead and paste that in here and it tells me how to get to that main.

So the first key is weather, the second key is 0,the third key is main.

So if I wanted to do that the first key is weather,second is zero,third is main.

And now when I hit save, my weatherDescription is going to be clear. So you can use this tool to make it much easier for you to get a value that is quite deep in the tree.

So this pressure for example exists under main and then pressure. But something like the icon will exist at weather 0 icon.

So as a challenge, I want you to create three variables inside here. So I'm going to delete everything we've got here and I want you to create three variables that are going to contain the temperature,so this value right here and then the condition,so this value right here, this is the condition number.

And then there's the city name right here,this one. Pause the video and try to create three variables that will contain the data that comes from our JSON response.

All right.

So first things first.

Let's try and create our temperature variable.

So I'm just going to use a var keyword for now

And I'm going to show you why very very soon.

So I'm going to create a temperature variable and it's going to be equal to jsonDecode and the source is going to be our data,and then I'm going to add some keys.

So our temperature is right here.

So I'm going to click on it,copy the part and put it down here to see how I need to get there.

So the first key is going to be called main and the second key is going to be called temp.

So this is my first variable. My second variable is going to be the condition.

And this is gonna be jsonDecode data and the condition comes from right here the weather ID.

So I'm going to copy the part here and paste it in here.

So for this one I have to access it through weather and then zero and then ID. For the final one, it's going to be the name.

So I'm going to call it cityName and it's going to be equal to again

jsonDecode, decode our data, and then the city name is down here.

So let's copy that path and let's see what that looks like.

OK.

So it's just a single key which is going to be the string name.

So now I have three variables that come from my JSON data and I've set them all to be variables.

So let's hit save. Now if we wanted to reduce the amount of repetition here and we actually want to simply decode the data just once,so we could create a new variable called decodedData and we could set it to equal jsonDecode data.

Then we can use decodedData in all three of these places.

And the reason why I'm setting it to a dynamic variable here,so not giving it a type outright, is because if we hit CONTROL + J or CONTROL + Q on this jsonDecode,then you can see the output is a dynamic type.

It doesn't know what it's going to be until the data gets processed.

So you can either leave everything with a type of dynamic by using the var keyword or you can look at the data and see what the data type could be.

So ID is probably going to be an integer,temperature is probably going to be a double and name is probably going to be a string. And you could change these to those particular types.

So we change this to a double, this to an int and this to a string.

So right now if I was to print out the temperature and condition and city name,then you can see that I don't actually have any problems here and there are no incompatibilities.

But similarly had I kept all of those as vars, even though it's less explicit, it still will work as a dynamic type. But if I get the type wrong, say I used a double instead of the int for our condition, then we'll get errors inside here.

Let's change this back to the right data types and I like having the right data type here instead of a var because it makes it more explicit. The only one that we should keep as dynamic is the one for our decodedData because we really won't know what the type is until the decoding has completed. Now so far, we've been learning about APIs and JSONs and how to decode our JSONs to actually get the values from the API response. But we've only been working with some dummy data from Open Weather Map because we're using samples.openweathermap.org rather than the actual API. And we're doing this because we don't yet have an app ID that will authorize us to use their API.

So in the next lesson, I'm going to show you how to register for your own app ID so that you can start parsing over the actual latitudes and longitudes that we're getting from our app to get the data for that location.

So all of that and more, I see on the next lesson.

